EXHIBIT 7

UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF MICHIGAN
SOUTHERN DIVISION

LEAGUE OF WOMEN VOTERS	)	
OF MICHIGAN, et al.,	)	Case No. 2:17-cv-14148
	)	
Plaintiffs,	)	Hon. Eric L. Clay
	)	Hon. Denise Page Hood
	)	Hon. Gordon J. Quist
v.	)	
	)	**DECLARATION OF**
	)	**CHARLES SEVERANCE**
RUTH JOHNSON, in her official	)	
Capacity as Michigan	)	
Secretary of State, et al.,	)	
	)	
Defendants.	)	

I, Charles Severance, declare under penalty of perjury and pursuant to 28 U.S.C. § 1746, as follows:

1. I am over 18 years of age and competent to testify.

2. I have personal knowledge of the matters set forth below.

3. I am a Clinical Professor of Information for the School of Information at the University of Michigan, Ann Arbor.

4. I am also a founding faculty member of the Informatics Concentration at the University of Michigan. Previously, I was the Executive Director of the Sakai Foundation and the Chief Architect of the Sakai Project, which is an open source learning management system that is composed of 2.5 million lines of Java. The Sakai

system is currently used at over 300 universities around the world. I am still involved in all aspects of the software design, development, testing, and release processes.

5. I have also written several books including, *Using the Google App Engine*, *Python for Informatics*, *High Performance Computing*, and *Sakai: Free as in Freedom*.

6. In 1985 I received my B.S. in computer science from Michigan State University. In 1990, I received my M.S. in computer science from Michigan State University. And in 1996, I received my Ph.D. in computer science from Michigan State University.

7. My research has been published in *Interactive Learning Environments Journal*, *Journal of Parallel and Distributed Computing*, and *O'Reilly and Associates*, which publishes information on computer technology. A true and accurate copy of my curriculum vitae is attached hereto as Exhibit A.

8. My academic areas of expertise include computer programming, high performance computing, informatics, and building learning management systems in a way that improves education.

9. I have particular expertise in Java, a type of computer programming language. I have taught several Java programming classes at the graduate level and been a professional developer of more than 30 Java projects over the past 15 years.

10. An individual may utilize Java to write source code, which is a file of readable code that is used to describe what the developer wants the program to

accomplish. A source code file is usually saved with a .java suffix. Source code is translatable into byte code through a compiler. Byte code consists of binary executable code and is typically saved with a .class suffix, and multiple .class files are combined into a file with a .jar suffix. Byte code can be translated back into source code through a process known as decompiling. Once it has been translated back, it is referred to as decompiled byte code.

11. You can write and/or modify Java source code with a wide range of file editing software.

12. When you make changes to a Java source code and save those changes to the computer hard drive, the updated code replaces the previous version of the source code.

13. In order to permanently save different versions of the source code, you may utilize web-based version control systems, such as GitHub, which have the capacity to simultaneously keep all of the versions of a file. Another approach is to save the updated file with a different file name on your personal hard drive.

14. Every time you compile byte code from source code, the resulting byte code is timestamped in the .jar file to reflect the date and time that it was compiled. Some compilers can read already compiled byte-code to produce a re-compiled version of that byte code. You may re-compile byte code for multiple reasons. One reason you may re-compile byte code is to shrink the code and reduce the size of the

byte code files. Every time you re-compile byte code, the re-compiled byte code is timestamped to reflect the date and time it was re-compiled. Re-compiling byte code does not change the underlying algorithm in the source code upon which the byte code is based.

15. To decompile byte code, you can use decompilers which are available online. Decompiled byte code is not as easy to read as source code. However, if significant expertise and resources are devoted to understanding the decompiled byte code, it can be done.

I declare under the penalty of perjury under the laws of the United States of America that the foregoing is all true and correct.

Date: December 13, 2018 ______________________________

Exhibit A

Charles Severance **(517) 694-0892** **c.severance@ieee.org**

Dr. Severance is a Clinical Assistant Professor at the University of Michigan School of Information.. He received Master's Degree in Computer Science from Michigan State University in 1990 and Phd. in Computer Science also from Michigan State University in 1996 in Computer Science. He is currently involved in online collaboration systems for teaching and learning as well as e-Research. Prior research interests include High Performance Computing, having written the O'Reilly book on the subject. He is the author of the O'Reilly book titled "Using the Google App Engine".

Education:
Ph.D., Michigan State University, 1996. Computer Science.
M.S., Michigan State University, 1990. Major Computer Science
B.S., Michigan State University, 1985. Major Computer Science

Work Experience:
09/07 – Present, University of Michigan School of Information, Clinical Assistant Professor.

08/02 – 08/07, University of Michigan Duderstadt Center, Executive Director Sakai Foundation (www.sakaiproject.org).

01/01 – 08/02, Director of Product Development, Strategic Interactive. Responsible for software design and development leadership.

06/99 – 01/01, Associate Director for Advanced Technology, University of Michigan Media Union. Responsibilities include Video/Audio Production Facilities, Scientific Computing, and Scientific Visualization.

04/96 – 05/99, Director of the Computer Services - MSU College of Engineering. Adjunct Assistant Professor, Department of Computer Science, Michigan State University.

10/86 – 03/96, Michigan State University - Computer Laboratory. Division Manager – UNIX Computing. Duties included project management, systems design, software development, system acquisition, new employee training, and employee evaluation.

06/83-10/85, Burroughs Corporation - Okemos, MI. Project Leader - Systems development group. Duties included project management, systems design, and employee evaluation.

06/78 – 06/83, Michigan State University - Computer Laboratory. Coordinated the general consulting activities of the User Information Center.

Selected Publications:

Severance, C, "Using Google App Engine," O'Reilly and Associates, May 2009, ISBN 978-0-596-80069-7.

Severance C, Hardin J, Whyte A, The Coming Functionality Mashup in Personal Learning Environments, Interactive Learning Environments (Journal), Volume 16, Number 1, April 2008.

Severance C, IEEE Computer, Standards Column Editor. Columns written: January 1999 - POSIX: A Model for the Future of Computing, July 1998 - Standardizing Real-Time Streaming Protocols, March 1998 - IEEE 754: An Interview with William Kahan, September 1997 - OSI Retrospect and Prospect, and February 1997 - Linking Computers and Consumer Electronics.

Severance C, Dowd K, "High Performance Computing", O'Reilly and Associates 1998. ISBN 1-56592-312-X.

Severance C, Enbody R, Petersen P, Managing the Overall Balance of Threads on a Multiprocessor using Automatic Self Allocating Threads (ASAT) - Journal of Parallel and Distributed Computing 37, 106-112 (1996).

Other Publications / Software Developed:
Sakai Project (www.sakaiproject.org) - I was the Chief Architect and later Executive Director of the Sakai Project.

Sync-O-Matic-2000, Web Lecture Capture using Real Video, www.syncomat.com

ClipBoard-2000, Web Lecture Capture using QuickTime, Copyright 1999-2003, www.netfact.com/csev/projects/cb2k/

CERN Web Lecture Archive, webcast.cern.ch/Projects/WebLectureArchive/ a joint project between CERN and University of Michigan

Synergistic Activities
Nuthin' but Net and Internet:TCI – Co-Host of two regional award-winning monthly televison talk shows about the Internet from 1995-1999.

Merit Network, Davidsen, S, et. al., "Building A Community Network Tool Kit", $177,200 1997. This project developed a book, training materials, video materials, software, and web site to support public libraries in their efforts to provide on-line community information (see www.citoolkit.org/).

“Collaborative Proposal: Middleware for Grid Portals”, National Science Foundation, $720,000 , 9/2003.